

Ruby - Bug #20736

prism emits wrong warnings in syntax-error code

09/13/2024 03:23 AM - mame (Yusuke Endoh)

Status:Feedback Priority:Normal Assignee:kddnewton (Kevin Newton) Target version: ruby -v:	Backport:3.1: UNKNOWN, 3.2: UNKNOWN, 3.3: UNKNOWN
Description <pre>\$./local/bin/ruby -w --parser=parse.y -e 'begin eval("0a"); rescue SyntaxError; end' \$./local/bin/ruby -w --parser=prism -e 'begin eval("0a"); rescue SyntaxError; end' (eval at -e:1):1: warning: possibly useless use of a literal in void context \$./local/bin/ruby -w --parser=parse.y -e 'begin eval("+a.0"); rescue SyntaxError; end' \$./local/bin/ruby -w --parser=prism -e 'begin eval("+a.0"); rescue SyntaxError; end' (eval at -e:1):1: warning: possibly useless use of +@ in void context</pre>	
Related issues: Related to Ruby - Bug #20817: Ruby 3.4.0dev emits `warning: possibly useless ... Rejected	

Associated revisions

Revision 76d927f88f9b64cf01912be41d5e88238a9e6ec6 - 09/17/2024 03:44 AM - mame (Yusuke Endoh)

Prevent prism warnings in syntax exhaustive tests

[Bug #20736]

Revision 76d927f88f9b64cf01912be41d5e88238a9e6ec6 - 09/17/2024 03:44 AM - mame (Yusuke Endoh)

Prevent prism warnings in syntax exhaustive tests

[Bug #20736]

Revision 76d927f8 - 09/17/2024 03:44 AM - mame (Yusuke Endoh)

Prevent prism warnings in syntax exhaustive tests

[Bug #20736]

History

#1 - 09/13/2024 03:32 PM - kddnewton (Kevin Newton)

This is because Prism recovers from the syntax error. For example:

```
eval("1; 2; 9a; 3; 4; 5; 9b")
```

with Prism this will give:

```
(eval at test.rb:1):1: warning: possibly useless use of a literal in void context
(eval at test.rb:1):1: warning: possibly useless use of a literal in void context
(eval at test.rb:1):1: warning: possibly useless use of a literal in void context
(eval at test.rb:1):1: warning: possibly useless use of a literal in void context
(eval at test.rb:1):1: warning: possibly useless use of a literal in void context
(eval at test.rb:1):1: warning: possibly useless use of a literal in void context
(eval at test.rb:1):1: warning: possibly useless use of a literal in void context
test.rb:1:in 'Kernel#eval': (eval at test.rb:1):1: syntax errors found (SyntaxError)
> 1 | 1; 2; 9a; 3; 4; 5; 9b
    |      ^ unexpected local variable or method, expecting end-of-input
    |      ^ unexpected local variable or method, expecting end-of-input

from test.rb:1:in '<main>'
```

with parse.y this will give:

```
(eval at test.rb:1):1: warning: possibly useless use of a literal in void context
```

```
(eval at test.rb:1):1: warning: possibly useless use of a literal in void context
test.rb:1:in 'Kernel#eval': (eval at test.rb:1):1: syntax error, unexpected local variable or method, expecting
end-of-input (SyntaxError)
1; 2; 9a; 3; 4; 5; 9b
    ^

from test.rb:1:in '<main>'
```

this is because parse.y is not recovering from this error. I would imagine when parse.y *does* recover from this error, it would have the same behavior.

So I'm not sure if this is wrong or not. [@mame \(Yusuke Endoh\)](#) what do you think?

#2 - 09/13/2024 05:02 PM - kddnewton (Kevin Newton)

- Status changed from Open to Feedback

#3 - 09/13/2024 07:51 PM - mame (Yusuke Endoh)

Although I can understand the technical reason, I still feel a bit uncomfortable when the warning says "use of a literal" since 9a is clearly not a literal to the human eye.

As a practical matter, this behavior causes a lot of warnings when you make test-all RUBYOPT=-w. Should we work around it by setting \$VERBOSE = nil?

```
[19134/33573] TestRubyLiteral#test_string(eval at /home/chkbuild/chkbuild/tmp/build/20240913T183003Z/ruby/test
/ruby/test_literal.rb:65):1: warning: possibly useless use of a literal in void context
(eval at /home/chkbuild/chkbuild/tmp/build/20240913T183003Z/ruby/test/ruby/test_literal.rb:68):1: warning: pos
sibly useless use of a literal in void context
(eval at /home/chkbuild/chkbuild/tmp/build/20240913T183003Z/ruby/test/ruby/test_literal.rb:71):1: warning: pos
sibly useless use of a literal in void context
(eval at /home/chkbuild/chkbuild/tmp/build/20240913T183003Z/ruby/test/ruby/test_literal.rb:74):1: warning: pos
sibly useless use of a literal in void context
...
```

#4 - 09/13/2024 08:58 PM - kddnewton (Kevin Newton)

Right now Prism is lexing 9a as an integer and then an identifier. So it's effectively parsing it as 1; 2; 9; a. That's why it's warning that 9 is a useless literal. I could change this in the parser to parse it instead as a missing . to make it a method call? Or whatever else you think makes more sense.

For make test-all yes, I would suggest turning off warnings, as if it's just meant for a syntax check you don't care about warnings in those cases.

#5 - 09/15/2024 12:46 AM - mame (Yusuke Endoh)

I wonder if error recovery is needed in the interpreter's parser. It might be a benefit to be able to output almost all syntax errors in one run. However, I am not sure about the benefit, given the such confusing warnings.

Well, let's stop the test warnings and try it for a while. If other problems arise, we should stop error recovery in the interpreter.

#6 - 09/15/2024 12:51 AM - mame (Yusuke Endoh)

<https://github.com/ruby/ruby/pull/11626>

#7 - 09/15/2024 10:25 AM - Eregon (Benoit Daloze)

I think good error recovery is key to have helpful error messages when writing Ruby code.

Maybe one way here is to not show warnings if there are any errors?

OTOH I think it is not a big usability problem, warnings appear first and the separation between warnings and errors is clear in the output. (and parse.y also shows both warnings and errors when there are both, as shown above)

However, I am not sure about the benefit, given the such confusing warnings.

These are very synthetic examples in test.

What matters is how helpful is the output for code (with errors) that users would write.

So I think setting \$VERBOSE to nil in these tests is the right fix.

It's also what is done e.g. [here](#).

If other problems arise, we should stop error recovery in the interpreter.

Let's not jump to such conclusions because some synthetic code in a test emits some confusing output, that code is confusing in the first place. Instead, I propose we evaluate based on code written by humans and not generated by tests.

#8 - 10/26/2024 08:23 AM - mame (Yusuke Endoh)

- Related to Bug #20817: Ruby 3.4.0dev emits `warning: possibly useless use of + in void context` while Ruby 3.3.5 does not added